Evaluation and Correction of Laser-Scanned Point Clouds

Christian Teutsch[*a] Tobias Isenberg[c] Erik Trostmann[a] Michael Weber[a]
Dirk Berndt[a] Thomas Strothotte[b]

[a]Dept. of Intelligent Sensor Systems, Fraunhofer IFF, Sandtorstr. 22, 39106 Magdeburg, Germany

[b]Dept. of Simulation and Graphics, Otto-v.-Guericke Univ., POB 4120, 39016 Magdeburg, Germany

[c]Dept. of Computer Science, Univ. of Calgary, 2500 Univ. Drive NW, Calgary, AB, Canada T2N 1N4

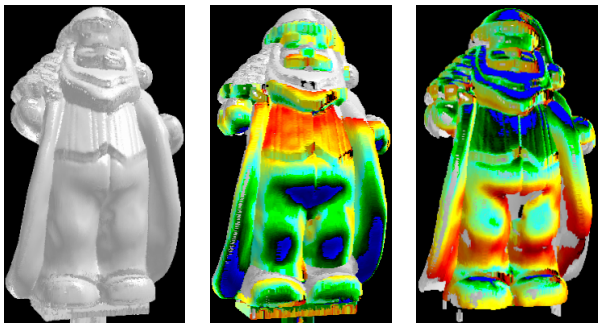
ABSTRACT

The digitalization of real-world objects is of great importance in various application domains. E. g.in industrial processes quality assurance is very important. Geometric properties of workpieces have to be measured. Traditionally, this is done with gauges which is somewhat subjective and time-consuming. We developed a robust optical laser scanner for the digitalization of arbitrary objects, primary, industrial workpieces. As measuring principle we use triangulation with structured lighting and a multi-axis locomotor system. Measurements on the generated data leads to incorrect results if the contained error is too high. Therefore, processes for geometric inspection under non-laboratory conditions are needed that are robust in permanent use and provide high accuracy as well as high operation speed. The many existing methods for polygonal mesh optimization produce very esthetic 3D models but often require user interaction and are limited in processing speed and/or accuracy. Furthermore, operations on optimized meshes consider the entire model and pay only little attention to individual measurements. However, many measurements contribute to parts or single scans with possibly strong differences between neighboring scans being lost during mesh construction. Also, most algorithms consider unsorted point clouds although the scanned data is structured through device properties and measuring principles. We use this underlying structure to achieve high processing speeds and extract intrinsic system parameters to use them for fast pre-processing.

Keywords: laser scan processing, point cloud analysis and evaluation, point cloud correction, B-spline approximation

1. INTRODUCTION

Geometric models are the very foundation of contemporary three-dimensional computer graphics. In addition to creating new models by using a modeling suite, the use of 3D scanning systems such as laser scanners has recently become more and more common. When reconstructing objects from laser scan data, usually very large data sets have to be processed. Therefore, it is often necessary to minimize the number of points while minimizing the loss of information at the same time. In addition, the generated point cloud usually contains a considerable number of errors. Most of these errors are directly dependent on the measurement system and the scanned object's surface. Outliers and other erroneous points are an important factor when discussing metering precision. Therefore, they have to be detected and removed from the point cloud or corrected in order to get a clean model that can be used as precise measuring data. Finally, an optimization of the laser scanning system can be realized based on the analysis of the data quality of each individual point in the data set (see Figure 1). This can be achieved by considering the specific settings of the scanning system, in particular, the parameters of the laser and the camera. Optimizing their positions based on the results of the point cloud analysis results in better point cloud quality in a subsequent scan.



(a) (b) (c)

Figure 1. Shaded point cloud from a model of Santa Claus (a). Quality of measured points depending on projection angles from a lower (b) and an upper (c) laser scanning sensor (increasing quality from red over yellow and green to blue).

*christian.teutsch@iff.fraunhofer.de, www.iff.fraunhofer.de

We consider two major steps: evaluating and optimizing the scanned data as well as improving the system's configuration. First, we present a fast pre-processing step that is integrated in the scanning process which processes each scanline individually. To detect and remove erroneous points we use, e. g.camera parameters. The laser position is used to sort the discontinued sublines that each scanline comprises. This sorted set of sublines is smoothed to minimize high frequent noise that was generated, for example, due to aliasing effects in the preceding image processing. Afterwards, scanlines are approximated by B-splines which are analyzed to extract, e. g.curvature. Thus, potential edges can be determined by analyzing the curvature of each point along a curve, too. This smoothing and approximation ensures that no additional error is introduced beyond measuring accuracy. Furthermore, we present methods also based on the system parameters and edge information that are well qualified for detecting simple geometrical primitives (e. g.circles, ellipses). In addition, scanlines are sorted based on the locomotor system's positions. The sorted set of scanlines is analyzed with respect to individual measurements and parameters that traditionally are determined within polygonal meshes (e. g.surface normals). For example, based on 2D/3D-viewing properties we determine quality values for each point of the point cloud. These are used for iteratively optimizing object, laser, and camera positions.

The major advantage of our procedures lies in that we do not consider and process a possibly erroneous mesh but instead work with the measured data directly–the point cloud. Compared to many algorithms that manipulate point clouds through an approximation with polygonal meshes, the goal of the algorithms here is to automatically correct each measurement individually and integrate the methods directly in the measurement process. Automation and industrial environments (e. g.dust, vibrations or changing lighting conditions) require a high degree of robustness. Therefore, compromises must be found between elegant and robust algorithms, that meet the individual requirements. The schematic processing of points clouds with our approaches is shown in Figure 2. These approaches and techniques are used to reconstruct a revised and

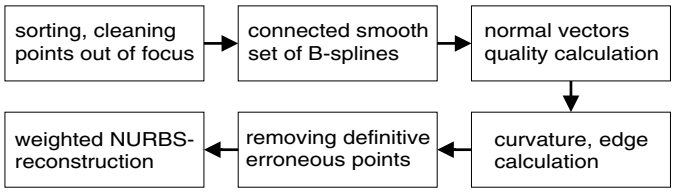


Figure 2. Schematic processing of point cloud data using our approaches.

optimized point cloud that is much better qualified for fast, robust, and precise measurements. Finally, we evaluate the effectiveness of the proposed methods based on exemplarily point cloud sets from various different models.

In summary, the main contributions of this paper are

- an analysis of the scanning process in which we identify several system parameters that can be used for a qualitative evaluation of the point cloud,
- various methods for a quantitative analysis of the data that are partially based on a B-spline approximation of the points in the point cloud,
- the detection of geometrical characteristics such as local curvature and edges on the surface of the object, and
- a method for reconstructing a revised and optimized point cloud based on these results.

In the remainder of this paper, we start by discussing related work on the area of 3D surveying and handling the obtained point clouds in Section 2. In Section 3, we analyze the used laser scanning system and its measuring principle in order to obtain potential sources of error. Based on this analysis, Section 4 derives methods for quantifying point clouds from system parameters. The obtained quality values are used in Section 5 to approximate adapted B-spline and NURBS curves in order to allow for optimizing the point cloud by thinning, smoothing, and closing gaps. While some intermediate results are given within these sections, in Section 6 we illustrate our methods with a number of example data sets and discuss the results. In Section 7, the proposed methods are summarized and some perspectives for extending our methods in the future are discussed.

2. RELATED WORK

3D scanning of complex shapes and capturing surface properties are typically a difficult task. Millions of points and corresponding information vectors are generated. There are several measuring systems with partially different principles, some commercial and some proprietary developments.

A well known example for scanning very large objects is *The Digital Michelangelo Project*.[1] LEVOY ET AL. describe a hardware and software system for digitizing the shape and color of large fragile objects under non-laboratory conditions using the example of the famous statues of Michelangelo. Another example is given by BERNARDINI ET AL. They describe a project to create a three-dimensional digital model of Michelangelo's Florentine Pietá.[2] A combined multi-view and photometric system is used to capture hundreds of small meshes on the surface. They present their methodology to acquire the data and construct a computer model of the large statue.[3] They also describe some preliminary studies being made by an art historian using the model.

The common way to process and analyze point clouds is to find (free-form) representations and use them as the basis for the following algorithms. Popular methods are triangulation algorithms such as *Marching Cubes* (first introduced by LORENSEN ET AL.[4]), *Delaunay Triangulation*,[5,6] and some shape descriptions based on B-spline and Bézier surfaces. HOPPE presents a modified version of the original marching cubes algorithm for triangulating iso-surfaces.[7] He constructs a triangle mesh by approximating a distance function over portions of the set of 3D data. In the next steps an energy function is minimized to get a regular and smooth mesh as surface description.[8] This method leads to smooth approximations but its accuracy cannot be controlled as it is needed, e. g.for precise geometric measurements. For a range scanner, NEUGEBAUER and KLEIN present a similar approach where the distance function is derived from the range images.[9] A triangulated model is generated from a rough approximation of the object using the marching cubes algorithm. Subsequently, the triangles representing the surface are adaptively subdivided until a pre-defined degree of accuracy has been reached.

As has been mentioned before, in most cases the scanned data set is erroneous. There are, for example, outlying points, gaps, and holes. These errors can lead to misarranged meshes. GROSSKOPF and NEUGEBAUER show the correction of partially incompletely captured surface portions through Geometrical Deformable Models (GDMs).[10] A GDM is defined as a triangle mesh that dynamically deforms by moving vertices based on forces. According to HOPPE ET AL.,[8] edges in a GDM act like springs and a pressure force is defined along the surface normals to cause a smooth deformation which results in a well conditioned polygonal mesh. NEUGEBAUER considers some effects that may appear when working with optical sensors.[11] When scanning a complete surface from different positions, the point cloud consists of several parts that are overlapping. NEUGEBAUER uses redundant areas to connect these partial point clouds and builds a mesh using marching cubes. BOEHLER ET AL. consider 3D scanning software for point cloud treatment with the goal to establish criteria for data cleaning and thinning.[12] On the basis of a description of several problems that can occur during the scanning process (e. g.reflections) they propose some basic techniques that satisfy their criteria.

For free-form topologies, ECK and HOPPE present a procedure for reconstructing a tensor product surface from a set of scanned 3D points.[13] They define a surface as a network of B-spline patches and explore adaptive refinement of the patch network in order to satisfy user-specified error tolerances. The main advantage of this method is that single patches represent local surface parts much better than global approximations. Thus, the influence of erroneous parts of a point cloud can be reduced. Complementary, PARK ET AL. present a method to represent an unorganized point set as a network of rational B-spline surfaces (NURBS).[14] This allows a more precise influence on the resulting surface because every single point can have a specific weight. Another approach given by DIETZ describes an entire surface with a single trimmed B-spline surface.[15] However, this method is only applicable for surfaces with low complexity. It typically uses existing triangle meshes as basis for a surface approximation. An extension to the approach is the work of BAJAJ ET AL.[16] They replace the planar faces of the Delaunay tetrahedron with weighted Bézier-patches to achieve well conditioned and esthetic surfaces.

Alternative methods of describing and visualizing point clouds are introduced by ALEXA ET AL.[17] Instead of connecting points to polygonal meshes they use the measured points itself to generate a surface. The so-called *point set surfaces* are generated by an internal transfer of the point set to a geometric description. With a local decomposition to Voronoi regions[6] an arbitrary amount of points is generated at positions of interest. Compared to polygonal meshes, the advantage of this method is that there is no explicit dependency between neighboring points. However, a closed surface visualization using this method requires a large number of points.

The main goal of the described methods is to compute smooth and esthetically pleasing shapes under some limitations. The feasibility of these methods for use in industrial measurement devices and applications depends on high processing speed and accuracy. However, a laser scanner normally produces large and dense point sets with erroneous points. Therefore, the methods described above can only partially be used for effective measurements.

3. DESCRIPTION OF THE LASER SCAN SYSTEM

For the demonstration of our approach we chose a laser scanning system that operates on the basis of triangulation with *structured lighting*. Structured lighting in this context is the projection of a light pattern (plane, grid, or a more complex shape) at a known angle onto an object. Using this principle, structured lighting can be used to determine the shape of objects in machine vision applications. In addition, it can be used to recognize and locate objects in an environment. These features make structured lighting very useful, for example, in assembly lines for implementing process or quality control. These processes use structured lighting for alignment and inspection purposes. Although other types of light can be used for structured lighting, laser light is usually the best choice because of its precision, intensity and reliability. However, our methods can easily be adapted to other well known measuring principles e. g.gray-code or phase-shift.[18] In order to identify parameters of the system that can be used for error analysis and point quality evaluation, we first give an overview of the measurement principles used in the examined laser scanning system.

3.1. Measuring Principle

The projection system of a laser scanner contains several sensors. Each sensor is a combination of at least one laser projection system that projects a light pattern onto the scanned objects and one camera that captures the projected image. The light pattern used most often in optical 3D scanners is generated by fanning out the light beam into a *sheet-of-light* meaning that a single laser beam is expanded by special lenses. When a sheet-of-light intersects an object, a bright line of light is projected onto the surface of the object. Through the topology of the surface this line is distorted. The camera observes this scene from an angle. By analyzing the line of light the observed distortions can be translated into height variations. Therefore, the positions of the laser and the camera are needed. They are derived from the system calibration. For each vertex in the 2D images, 3D point coordinates are computed using the parameters from the calibration. Hence, structured lighting is sometimes described as *active triangulation*. Through a process of rotating and translating the object the whole surface can be mapped into 3D coordinates. The resulting point cloud consists of a set of single deformed lines, each line (curve) representing the results from a single sheet-of-light measuring.

The scanning system consists of several parts for hardware and software processing. In the preliminary 2D image, a processing step detects and analyzes the projected laser line. Due to the pixel raster in a digital image, aliasing effects may occur. These effects lead to high frequency noise in the resulting 3D point cloud. Therefore, each measured scanline is smoothed (see Section 5.2) to reduce these effects. In addition, each measure contains noise and other artifacts such as gaps and holes due to shadowing effects.

It is important to reliably estimate the uncertainty of the measurement for being able to develop algorithms that can guarantee a certain accuracy. The first step is to specify and determine the system's parameter. In the examined laser scanning system, the following parameters could be identified. Due to the system architecture and the measuring process, the scans result in a sorted set of scanlines with sublines. The location of the projection system is known, i. e., the positions of the laser and the camera. Since the camera observes the object it is possible to estimate the surface normal of each surface point depending on its neighbors. One parameter that also influences the system is the locomotor system, which moves the object relatively to the sensors and has a certain error tolerance, too. The system we developed is designed to scan objects in a volume of $400x400x400\,mm^3$ with a measuring uncertainty of 20–100 μm depending on the surface properties. Typically, a set of 6–10 million points of the objects surface are digitized.

3.2. Characterization of Errors

There are many sources for erroneous scan data. The most severe cases will be discussed below. As can be deduced from numerous example scans, most of the outliers and other erroneous points are caused by reflections. In these cases, the high energy laser beam is reflected from mirroring surfaces such as metal or glass. Therefore, too much light hits the sensor of the camera and so-called *blooming effects*[*] occur. In other cases, also diffuse reflections may miss the camera. In addition,

[*] If too much light hits the sensor of a CCD camera, some cells cannot hold the generated energy and it is distributed to neighboring cells. This is called a blooming effect.

a part of the object may lie in the path from the projected laser line to the camera causing a *shadowing effect*. All these effects are responsible for gaps and holes. At sharp edges of some objects, partial reflections appear. In addition, craggy surfaces cause multiple reflections and, therefore, indefinite point correlations. Other problems are caused by possible range differences originating from systematic range errors resulting from different reflectivity of the surfaces elements. Since the scanner systems are typically used in industrial environments, some atmospheric effects (e. g.dust) may affect the quality of the image obtained by the camera. Furthermore, aliasing effects in the 2D image processing and *speckle effects* lead to high frequent noise in the generated 3D data.

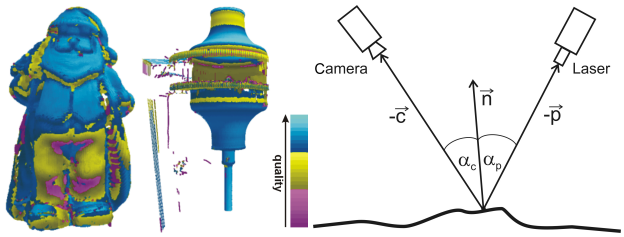
Therefore, the resulting 3D point data is noisy and partially erroneous. However, a lot of these errors can be minimized by an optimal alignment of the projection system and the object surface so that the number of reflections is as low as possible. In order to arrange the setup properly, the quality of the generated point cloud has to be analyzed and evaluated.

4. POINT CLOUD QUALITY MEASUREMENT

The first step for optimizing a point cloud is to optimize the projection and viewing conditions. Therefore, the quality of the point cloud has to be quantified with respect to the position of laser and camera. Improving the recording conditions leads to less erroneous 3D points. Afterwards, the remaining errors can be detected and evaluated much easier.

4.1. Quality Measurement Using 3D Data

Because the laser scan process uses optical sensors, the quality directly depends on the viewing and projection properties. The smaller the angle between surface normal and direction of projection or viewing, α_p and α_c respectively, the better the surface was seen (see Figure 3). In addition, the triangulation between projection vector $\overrightarrow{p}$ and the camera viewing vector $\overrightarrow{c}$ is optimal when the angle $(\alpha_p + \alpha_c)$ defined by them is 90°. With this constraint there are less intersecting errors (e. g.convergent rays) and less numerical errors when computing the location of the surface point. The more the angle $(\alpha_p + \alpha_c)$ deviates from 90° and the larger the angles themselves the worse are the viewing conditions of the surface point for camera and/or laser.



(a) Santa and Converter. (b) Projection angles.

Figure 3. Quality of several measurements (a) depending on viewing and projection angles (b).

For the calculation of the point quality, the surface normals have to be known. In general, they can be derived from a triangle mesh. However, for our approach a polygonal mesh is not needed. Over a defined neighborhood for each point, a plane is approximated and its normal is used as the normal for the considered point. The neighborhood can be determined very fast since the sorting of the scanlines is a known system parameter. To guarantee a consistent normal orientation, the camera position is used as well. Obviously, the surface normal has to point towards the camera that recorded this point. Therefore, the correct surface normal orientation can be computed by calculating the angle between normal vector $\overrightarrow{n}$ and viewing vector $\overrightarrow{c}$. The normal is oriented correctly if this is $\leq 90^\circ$.

Hence, an important step to optimize a laser scan system is to perform an initial scan and evaluate the result based on the position and alignment of the projection system. Depending on the result, the viewing conditions of the object are optimized. In addition, for each point a corresponding quality for the viewing condition can be computed and used for further error compensation. Besides the quality in 3D, each surface point can additionally be evaluated by quantifying its quality in the 2D image processing. The intermediary results are numbers, scaled between 0 and 1. The limits in 3D are implicitly given by the worst and best possible projection. In 2D the limits are defined by empiric and statistic tests (e. g.contrast).

4.2. Quality Measurement Using 2D Data

In addition to the previously discussed viewing conditions, there are factors in the image processing step that may influence the resulting point cloud. These factors are influenced by environmental effects such as lighting, laser light energy, surface type, and camera resolution. For example, a high contrast of a laser line in the recorded image results in a high stability and precise detection. In addition, the line thickness is an important parameter. Thick lines make it harder to find its

exact middle and small features may be buried. Therefore, the line profile at each vertex has to be considered in the point quantification as well. The higher the slope of an edge orthogonal to the line the better its middle can be detected as a maximum in the profile.

4.3. Additional Quality Estimation

In our experiments we found that erroneous sublines have specific characteristics. On the one hand, they are very short with many high curvature points. High curvatures can be detected by analyzing the approximated curves. On the other hand, sublines may be caused by reflections. These lines are projected somewhere in space and, therefore, have no neighboring scanlines. These aspects are used for quantification as well.

5. AUTOMATED POINT CLOUD CORRECTION

After having discussed methods for evaluating point clouds, we now describe the general approach for correcting single scanlines. Measurements and methods of visualization (e. g.triangulation) do not require hundreds of thousands or even millions of points and large data sets need a lot of computation time and memory. Precise measurements, on the other hand, need data that is free of noise that may be caused by external effects during a scan. Therefore, the goal is to minimize the number of points and clean the point clouds from noise. On the basis of the measuring principle and the system parameters we now describe fast and effective methods for processing large point sets derived from a structured light scanning system. The resulting point cloud consists of a number of individual measurements (scanlines). Therefore, it is reasonable to describe the whole point cloud as set of sorted lines or curves to obtain a mathematical description. However, a set of local approximations is not sufficient for a evaluation of global neighborhoods. Therefore, the positions of the locomotor systems are used because the scanlines are sorted implicitly by sequence of the scans since the object is moved in front of the sensors.

The basic problem for the approximation is the choice of the mathematical description. One possibility is to use polynomial functions of a certain degree n and a set of parameters p_i in the form of:

$$f(x) = \sum_{i=0}^{n} p_i x^i \,. \tag{1}$$

In general, n different points can be described by a unique polynomial of degree $n-1$. Typical methods for polynomial interpolation are given by LAGRANGE, NEVILLE[19] or the least-squares method. Because of the large data sets, the unique polynomial is from a corresponding degree. With respect to computation time and numerical stability, the calculation of such a polynomial in $\mathbb{R}^3$ is not reasonable. Furthermore, each single point has the same influence on the resulting function. Thus, this kind of approximation is not robust enough against outliers. After all, functions in the form of $y = f(x)$ or $z = f(x,y)$ are depending on the axes of the coordinate systems and are, thus, not flexible. Therefore, it is reasonable to describe curves in $\mathbb{R}^3$ with parametric functions in the form of $\overrightarrow{r}(t) = [x(t), y(t), z(t)]$. To minimize the influence of single erroneous points, the whole curve should consist of local, single connected curves with low degree. The sum of the conditions is satisfied by B-spline curves. Hence, for our purpose each subline of one discontinuous measured scanline is processed and approximated by a B-spline curve.

5.1. B-Spline Approximation

For analyzing a set of points on a curve, B-splines are helpful. They are used to obtain a mathematical description from which features can be easily extracted. In addition, they can be used to close small gaps between neighboring sublines. Interpolating these gaps keeps the precision of measurements if the distance between the corresponding sublines is less than a certain threshold (e. g.5 mm; this depends on the accepted inaccuracy of the closing segment). Otherwise, the interpolated spline would follow the assumed geometry insufficiently.

A B-spline curve $x(t)$ of order k (degree $k-1$) is defined over an ordered knot vector T as vectorial polynomial:

$$x(t) = \sum_{i=0}^{n} \mathbf{d}_i N_{i,k,T}(t), \qquad t \in [t_{k-1}, t_{n+1}] \tag{2}$$

with the basis functions $N_{i,k,T}(t)$ and the control points $\mathbf{d}_i$.[20] These B-spline curves offer C^{k-2} continuity at the segment transitions.

To ensure that a B-spline curve approximates given points in an optimal way, control points have to be generated from the measured points. Therefore, the distance between the points on the curve X_i and the measured points M_i has to be minimized. A well-known and fast but relatively inexact method to achieve this is to minimize the quadratic (Euclidean) distance:

$$\sum_{i=1}^{n} \|X_i - M_i\|^2 = min\,. \tag{3}$$

One way to solve this problem is to use the least-squares-method. Control points D are computed depending on the values of the basis functions N and the measured points M:

$$\mathbf{D} = ((\mathbf{N}^T \cdot \mathbf{N})^{-1} \cdot \mathbf{N}^T) \cdot \mathbf{M}\,. \tag{4}$$

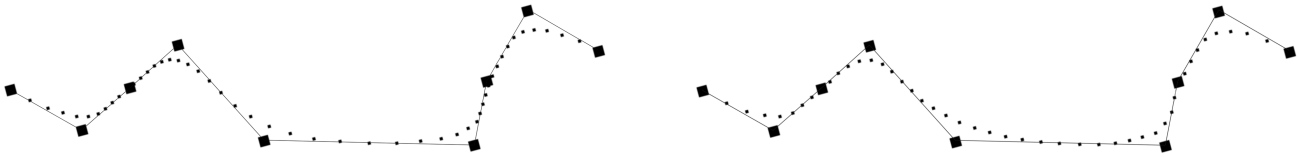
An approach for solving this problem using different distance norms as a linear programming problem is presented by HEIDRICH ET AL.[21]

5.2. Manipulating the B-spline Representations of Sublines

For further processing, it is useful to generate regularly spaced points. Therefore, the values of the parameter t of the B-spline curve have to be adapted. This can be achieved by choosing a knot vector with parameter distances that are proportional to that of the control points (chord distances) using the following ratio:

$$\frac{t_{i+1} - t_i}{t_{i+2} - t_{i+1}} = \frac{\|d_{i+1} - d_i\|}{\|d_{i+2} - d_{i+1}\|}\,. \tag{5}$$

The effect of this non-uniform parameterization is shown in 4(b). The resulting points on the curve are distributed more regularly and represent the geometry better.



(a) Irregular point distribution of a B-spline with uniform knot vector.

(b) More regular point distribution of a B-spline with non-uniform knot vector.

Figure 4. B-Spline approximation with different knot vectors resulting in different point distributions on the curves.

This approach is also useful to vary the point density on the approximated curve. For example, by calculating 10 regularly spaced points on a curve with an arc length of 10 mm a point density of 1 mm is achieved. For performance reasons, the real arc length s (see Eq.6) is estimated as the sum of all distances between neighboring points.

$$s = \int_{t_1}^{t+n+k+1} \sqrt{x'(t)^2 + y'(t)^2 + z'(t)^2}\,. \tag{6}$$

Through aliasing effects in the 2D image processing step, high frequent noise was generated. Therefore, the data has to be smoothed using, e. g.the approximated B-spline curves. HADENFELD presents an iterative method for smoothing B-splines.[22] This method is based on minimizing the bending energy of a thin, elastic bar with a constant cross-section. The corresponding term for a B-spline $x(t)$ is as follows:

$$E = \int (\ddot{x}(t))^2 \quad dt = min\,. \tag{7}$$

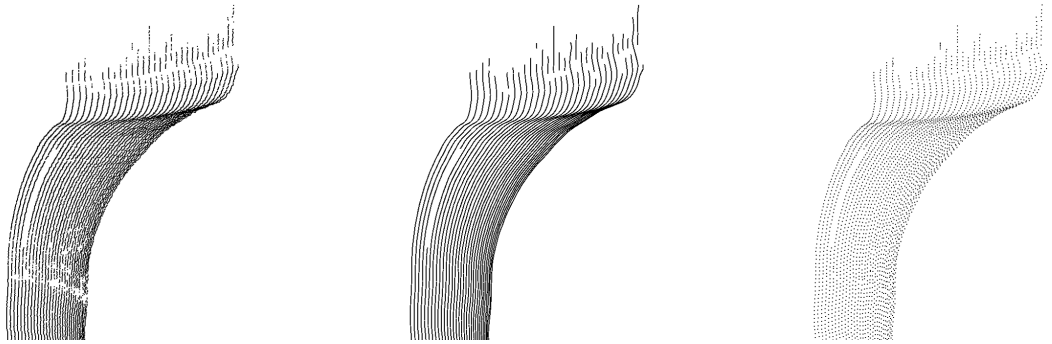
Each considered control point of B-spline $\tilde{d_i}$ is manipulated depending on the control points $\bar{d_i}$ from the last iteration:

$$\tilde{d_i} = -\frac{1}{16}\bar{d}_{i-3} + \frac{9}{16}\bar{d}_{i-1} + \frac{9}{16}\bar{d}_{i+1} - \frac{1}{16}\bar{d}_{i+3}\,. \tag{8}$$

Furthermore, a tolerance δ is added to limit the movement of control points with respect to the metering precision. Within each iteration the tolerance is checked with:

$$\tilde{d}_r^* = \begin{cases} \tilde{d}_i, & \text{if } \|d_i - \tilde{d}_i\| \leq \delta \\ d_i + \delta \cdot \frac{\tilde{d}_i - d_i}{\|\tilde{d}_i - d_i\|}, & \text{if } \|d_i - \tilde{d}_i\| > \delta. \end{cases} \tag{9}$$

The results of smoothing and thinning an exemplary point cloud with the help of B-splines are illustrated in Figure 5.



(a) Original point cloud segment.

(b) Closed and smoothed B-spline approximation.

(c) Finally smoothed and thinned segment.

Figure 5. Steps for smoothing and thinning using B-splines.

ECK and HADENFELD describe a method for removing control points of a B-spline curve.[23] Their algorithm implicitly detects redundant control points, which are not needed for an exact reconstruction. Furthermore, they minimize the error when adding or removing control points in general. SCHUMAKER and STANLEY present another method for a shape-preserving knot removal with respect to a given tolerance.[24] This method is very fast but it is used for quadratic splines only.

5.3. Detection and Deletion of Erroneous Points

A straightforward approach for minimizing the number of points and correcting the point cloud is to test for the focus space. All parts of a scanned point cloud that lie not within this area obviously should not be used, therefore, they can be deleted. This can be problematic if parts of a subline lie within this area and other parts do not. However, we found that such sublines should not be divided and the potential error can be accepted. In addition, a simple test can be used for sublines that consist of only a few points. In our test, sublines that had no neighbors and consisted of less than 10 points could safely be assumed to be errors.

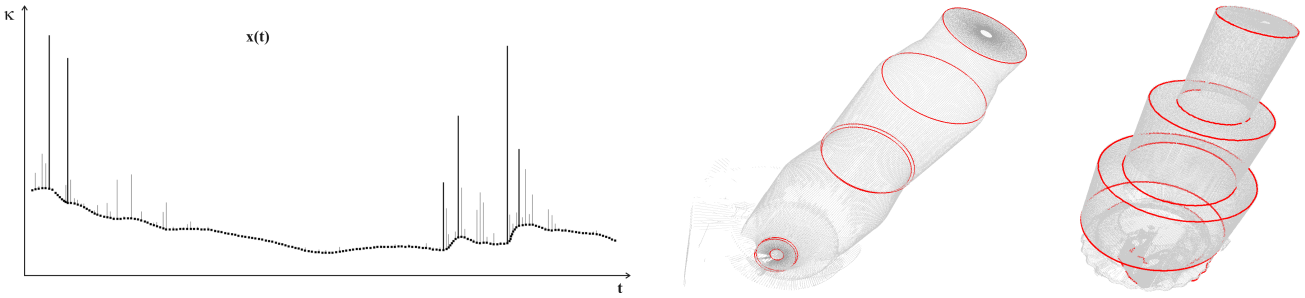
5.4. Analyzing Curvature Patterns

Most of the erroneous sublines exhibit conspicuous curvature patterns. For example, within short distances lots of turn-arounds and sharp edges occur indicated by high curvatures. In general, the curvature of parameterized curves can be calculated using:

$$\kappa = \frac{\|\vec{r}'(t) \times \vec{r}''(t)\|}{\|\vec{r}'(t)\|^3}. \tag{10}$$

In our analysis we found that curvatures of $\kappa > 0.2$ indicate sharp edges reliably (4^{th} order B-Splines). These values can be used to detect potentially erroneous sublines. In many tests with different kinds of objects and curves it was found that sublines with at least 40% of their points having a curvature $\kappa > 0.2$ can considered to be errors and can be deleted. The mentioned parameters are stable in a range of $\pm 10\%$, depending on the surface properties, noise, etc.

Sharp edges partially describe the shape of an object. Thus, in addition to use this information for error detection, it can also be very helpful for automatic object recognition (see Figure 6(b)). Further triangulation algorithms could use this edge information for an improved triangle mesh.



(a) Curvatures on a B-spline curve. (b) Derived object parts. (c) Derived object parts.

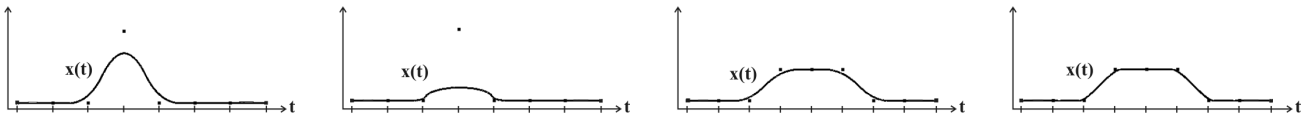
Figure 6. Identification of object structures using curvature (e. g.circles (b), circular parts (c), ellipses); $\kappa > 0.2$ in red.

However, the curvature can also be used to control the interpolated geometry. While B-splines smooth sharp edges, curvature can be used to correct the curve at the edges. In addition, the weights from the evaluation of the 2D and 3D data can be used to manipulate the path of the curve depending on the quality of each vertex in the image and point in the point cloud. Therefore, rational curves are needed for weighting single points.

5.5. Reconstruction Using NURBS Curves

The calculation of NURBS curves is similar to that of B-splines. In addition, for each control point d_i a weight h_i can be specified (see Equation 11). This weight determines the influence of the control point on the curve's path. It can be computed based on each point's quality (see Figure 7). Additional descriptions may be used as well.[20, 25]

$$x(t) = \frac{\sum_{i=0}^{n} d_i h_i N_{i,k,T}(t)}{\sum_{i=0}^{n} h_i N_{i,k,T}(t)}, \qquad t \in [t_{k-1}, t_{n+1}] \tag{11}$$



(a) Original. (b) Weighting outliers. (c) Original. (d) Weighting edges.

Figure 7. Two examples for correcting a scanline using NURBS.

To reconstruct an optimized scanline and, thus, an optimized point cloud, we propose the following rules to weight the control points depending on the curvature properties and the normalized quality values from the quantification (see Section 4):

$$W = \begin{cases} 1 & \text{for all not rateable points} \\ 1 + w_c + w_l + 0.5 w_t + 1 & \text{for } \kappa > 0,2 \\ 1 + w_c + w_l + 0.5 w_t + \kappa & \text{otherwise} \end{cases} . \tag{12}$$

w_c, w_l - weights for camera viewing angle and laser projection angle.
w_t - weight for triangulation angle (lower due to its relationship to w_l and w_c).

6. CASE STUDIES

The effectiveness of our approaches strongly depends on the properties of the surface to be measured (recall Section 3.2). For this reason, we tested our methods on different point clouds captured from different surface types and surface topologies. For the demonstration of results Figure 8 on the last page shows four phases of our approach. For a better visibility

of the manipulations made, each first image per row shows a simple and non-optimized triangulation of the raw, erroneous data. The second image illustrates the quality of each measured point miscolored for one sensor (laser and camera). Significant curvatures respectively object structures used for the NURBS-approximation are displayed in the third image. After processing the whole data with the presented methods, the obtained optimized point cloud is shown again as polygonal representation in the fourth image. For demonstration we chose a technical element (catalytic converter for vehicles), a plaster model of Santa Claus, a ceramic model of a duck and a green pepper as an organic object.

The metal, and thus reflective, surface of the converter caused a lot of errors that could largely be detected and corrected (see Figures 8(a) and 8(d)). In particular, the originally craggy scan of the surface of the converter could be smoothed by taking the estimated metering precision of the whole system into account. Considering the model of Santa Claus, the plaster surface caused only few artifacts that largely could be removed. However, the noise in the areas of his face and his legs could be notably reduced (see Figures 8(e) and 8(h)). The overall quality of the 3D data from the duck could be increased, especially to be seen at the reflections on its sides. The surface of the green pepper partly was translucent. By adapting the laser energy, a good measurement was possible. Nevertheless, the measured data was noisy but could sufficiently be smoothed. Attention should be paid to the parameters of the B-spline approximation (see Section 5.1). Closing small gaps can improve the consistency of the surface but real existing gaps were closed too (see holes in the object carrier in Figures 8(l) and 8(p)).

The proposed methods are stable, the mostly correct surfaces are not changed substantially. In contrast, the applied modifications are all within the error tolerance of the laser scanning system. In addition, the density of example point clouds could be reduced on average by 60% without appreciable loss of information.

7. SUMMARY AND FUTURE WORK

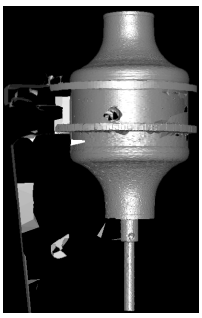
In this paper we presented methods to evaluate, quantify, and correct point clouds generated by an optical 3D scanning system automatically. We proposed techniques based on the system's parameters in 2D (e. g.contrast and line thickness) and in 3D (e. g.camera and laser positions, focus area, etc.) to estimate the quality of each single point. Furthermore, we approximated point clouds by a sorted set of B-spline curves for iterative smoothing and closing gaps. We derived edge information from these curves and reconstructed scanlines by using NURBS curves with respect to quality and curvature of each single point on it. The adjustment made to the point clouds can be controlled by tolerances so the metering precision of the whole system is not declined (removing points does not influence the precision). The main goal of our considerations was to analyze and correct the 3D point sets. One additional result is that most of the presented algorithms can directly be integrated in the preliminary 2D image processing step which results in drastically increasing processing speed (up to 40%, depending on the number of points). The proposed methods are based on a general optical measuring principle that is also used in other scanning systems and can, thus, easily be adapted to other scanning systems.

The described methods and the acquired systematic coherences allow further considerations of how to use the obtained parameters. The quality values from the point cloud quantification can be used for an automated object and system device positioning to improve the general scanning properties. This can be realized with an iterative process by measuring and evaluating until an optimum is reached. Furthermore, triangulation algorithms can use the smoothed and optimized point set as well as the edge information and the computed surface normals for more robust meshing. In addition, the edge information can be used for an automated object recognition. Finally, the developed algorithms can be extended to NURBS and B-spline surfaces to allow more detailed interrelated considerations and interpolations on neighboring scanlines.

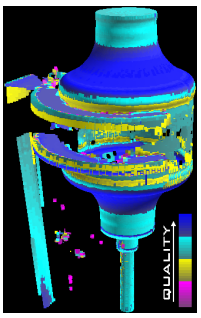
REFERENCES

1. M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The Digital Michelangelo Project: 3D Scanning of Large Statues," in *Proc. SIGGRAPH 2000*, K. Akeley, ed., pp. 131–144, ACM SIGGRAPH, Addison Wesley, 2000.
2. F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin, "Building a Digital Model of Michelangelo's Florentine Pietà," *IEEE Computer Graphics & Applications* **22**, pp. 59–67, Jan. 2002.
3. F. Bernardini and H. Rushmeier, "The 3D model acquisition pipeline," *Computer Graphics Forum* **21**, pp. 149–172, June 2002.
4. W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics* **21**, pp. 163–169, July 1987.

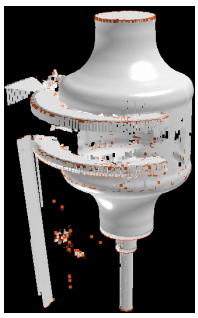
5. B. N. Delaunay, "Sur la sphere vide," in *Proceedings of the International Congress of Mathematicians, (Toronto, Canada, August 11–16, 1924)*, pp. 695–700, University of Toronto Press (1928), 1924.
6. F. Aurenhammer, "Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys* **23**, pp. 345–405, Sept. 1991.
7. H. Hoppe, *Surface Reconstruction from Unorganized Points*. PhD thesis, University of Washington, 1994.
8. H. Hoppe, T. DeRose, T. Duchamp, J. McDondald, and W. Stuetzle, "Mesh Optimization," in *Proc. SIGGRAPH 93*, pp. 19–26, ACM Press, (New York), 1993.
9. P. J. Neugebauer and K. Klein, "Adaptive Triangulation of Objects Reconstructed from Multiple Range Imgages," in *Proc. of IEEE Visualization 97*, pp. 41–44, ACM Press, 1997.
10. S. Grosskopf and P. J. Neugebauer, "Fitting Geometrical Deformable Models to Registered Range Images," in *3D Structure from Multiple Images of Large-Scale Environments, European Workshop, SMILE'98, Freiburg, Germany, June 6–7, 1998*, R. Koch and L. J. V. Gool, eds., *Lecture Notes in Computer Science* **1506**, pp. 266–274, Springer, (Berlin), 1998.
11. P. J. Neugebauer, "Scanning and Reconstruction of Work Pieces from Range Images," in *Second IFIP 5.10 Workshop on Virtual Prototyping (Arlington, Texas, May 6–8, 1996)*, Automation and Robotics Research Institute, 1996.
12. W. Boehler, G. Heinz, A. Marbs, and M. Siebold, "3D Scanning Software: An Introduction," in *Proc. of Int. Workshop on Scanning for Cultural Heritage Recording (Corfu, Greece, September 1–2, 2002)*, pp. 42–47, International Society for Photogrammetry and Remote Sensing, 2002.
13. M. Eck and H. Hoppe, "Automatic Reconstruction of B-spline Surfaces of Arbitrary Topological Type," *Computer Graphics* **30**, pp. 325–334, Aug. 1996.
14. I. K. Park, I. D. Yun, and S. U. Lee, "Constructing nurbs surface model from scattered and unorganized range data," in *Proceedings of the 2nd International Conference on 3-D Imaging and Modeling (Ottawa, Canada, October 04–08, 1999), Annual Conference Proceedings*, pp. 312–340, National Research Council of Canada, 1999.
15. U. Dietz, *Geometrie-Rekonstruktion aus Messpunktwolken mit glatten B-Spline-Flächen*. PhD thesis, Technische Universität Darmstadt, 1998.
16. C. L. Bajaj, F. Bernardini, and G. Xu, "Reconstructing surfaces and functions on surfaces from unorganized three-dimensional data," *Algorithmica* **19**, pp. 243–261, Sept. 1997.
17. M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," *IEEE Visualization 2001* , pp. 21–28, Oct. 2001.
18. R. Zumbrunn, "Automated fast shape determination of diffuse reflecting objects at close range by means of structured light and digital phase measurement," in *Proceedings of ISPRS Intermission Conference on Fast Processing of Photogrammetric Data, (Interlaken, Switzerland, June 2–4, 1987), Conference Proceedings*, pp. 12.1–12.9, 1987.
19. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 2. ed., 1999.
20. L. Piegl and W. Tiller, *The NURBS Book*, Monographs in Visual Communication, Springer, Berlin, 1995.
21. W. Heidrich, R. Bartels, and G. Labahn, "Fitting Uncertain Data with NURBS," in *Proc. 3rd Int. Conf. on Curves and Surfaces in Geometric Design*, pp. 1–8, Vanderbilt University Press, (Nashville), 1996.
22. J. Hadenfeld, *Iteratives Glätten von B-Spline Kurven und B-Spline Flächen*. PhD thesis, Technische Universität Darmstadt, 1998.
23. M. Eck and J. Hadenfeld, "Knot removal for B-spline curves," *Computer Aided Geometric Design* **12**(3), pp. 259–282, 1995.
24. L. L. Schumaker and S. S. Stanley, "Shape-preserving knot removal," *Computer Aided Geometric Design* **13**(9), pp. 851–872, 1996.
25. D. F. Rogers, *An Introduction to NURBS*, Academic Press, San Diego, 2001.



(a) Original.



(b) Evaluated.



(c) Significant curvatures.

(d) Processed.

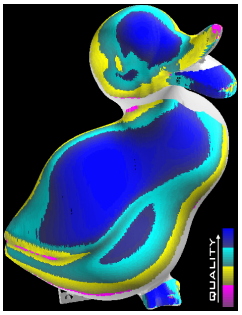
(e) Original.

(f) Evaluated.

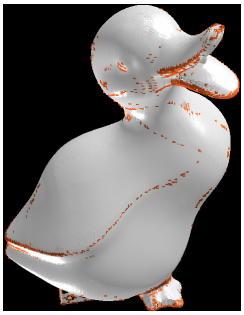
(g) Significant curvatures.

(h) Processed.

(i) Original.



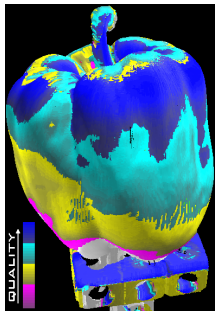
(j) Evaluated.



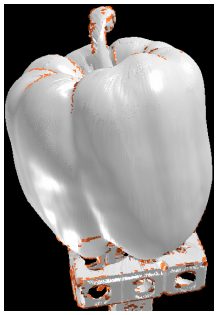
(k) Significant curvatures.

(l) Processed.

(m) Original.



(n) Evaluated.



(o) Significant curvatures.

(p) Processed.

Figure 8. Examples for triangulated point clouds, before and after processing. For each row: triangulated raw data, acquired quality, rough edges, triangulated optimized data.